



TECHNICKÁ UNIVERZITA V LIBERCI
Ekonomická fakulta



OVĚŘOVÁNÍ A CERTIFIKACE ŽIVOTNÍHO CYKLU SOFTWARE PODLE NORMY ISO/IEC 12207

Bakalářská práce

Studijní program: B6209 – Systémové inženýrství a informatika

Studijní obor: 6209R021 – Manažerská informatika

Autor práce: **Jiří Janovec**

Vedoucí práce: Ing. David Kubát





VERIFICATION AND CERTIFICATION OF SOFTWARE LIFE CYCLE ACCORDING TO ISO/IEC 12207

Bachelor thesis

Study programme: B6209 – System Engineering and Informatics

Study branch: 6209R021 – Managerial Informatics

Author: **Jiří Janovec**

Supervisor: Ing. David Kubát



Tento list nahradte
originálem zadání.

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Anotace

Tématem této práce je certifikace softwaru z hlediska procesů souvisejících s jeho životními cykly dle normy ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.

V první části práce se věnuji normalizaci jako takové a pojmům s ní souvisejícími, různým životním cyklům softwaru a definicím samotné normy ISO/IEC 12007:2008. V praktické části se pak zaměřuji na tvorbu metodiky/check-listu využitelného pro potřeby certifikace dle této normy certifikačním orgánem EZÚ, s.p.

Klíčová slova:

software, certifikace, ISO/IEC, životní cyklus

Anotation

Topic of this thesis is certification of software based on his life cycle processes in accordance with ISO/IEC 12207:2008 Systems and Software engineering – Software life cycle processes.

In the first part I am writing about normalization and about terms around it, different life cycles of software and about definitions of the ISO/IES 12207:2008 itself. In the practical part I am focusing on the creation of metodics/check-list usable for certification in accordance with this standard by the certifying authority of EZÚ, s.p.

Keywords: software, certification, ISO/IEC, life cycle

Obsah

Seznam obrázků.....	- 9 -
1 Rešerše.....	- 10 -
2 Základní pojmy.....	- 12 -
3 Životní cyklus softwaru	- 18 -
3.1 Vodopádový model.....	- 18 -
3.2 Prototypový model.....	- 21 -
3.3 Spirálový model.....	- 22 -
3.4 Inkrementální model	- 24 -
3.5 RAD.....	- 24 -
3.6 Agilní přístupy	- 25 -
4 Norma ISO/IEC 12207:2008.....	- 27 -
5 Praktická část.....	- 30 -
5.1 Vstupní nástroje auditora	- 30 -
5.1.1 Zkušenosti auditora	- 30 -
5.1.2 Plán auditu	- 31 -
5.1.3 Check list	- 32 -
5.2 Rozdělení procesů.....	- 33 -
5.2.1 Smluvní procesy	- 34 -
5.2.2 Procesy přípravné fáze projektů	- 34 -
5.2.3 Projektové procesy	- 34 -
5.2.4 Technické procesy	- 35 -
5.2.5 Procesy implementace softwaru	- 35 -
5.2.6 Podpůrné procesy	- 35 -
5.2.7 Procesy opětovného využití.....	- 36 -
5.3 Vytvoření struktury.....	- 36 -
5.4 Analýza výstupů procesů	- 37 -
5.5 Finalizace	- 47 -
5.6 Vyhodnocení zpracovaného řešení	- 48 -
6 Závěr.....	- 49 -
7 Seznam citací.....	- 50 -
8 Bibliografie	- 53 -
9 Seznam příloh.....	- 54 -

Seznam obrázků

Obrázek č. 1 – Vodopádový model

Obrázek č. 2 – Prototypový model

Obrázek č. 3 – Rozdělení procesů normy ISO/IEC 12207:2008

Obrázek č. 4 – Ukázka návrhu dokumentu založená na ČSN EN ISO 9001:2008

Obrázek č. 5 – Ukázka plánu auditu

Obrázek č. 6 – Záhloví check listu

1 Rešerše

První zmínka o fázích vývoje softwaru, které vzdáleně připomínaly pozdější vodopádový model, padla na konferenci Symposium on advanced programming methods for digital computers 29. června 1956 z úst Herberta D. Beningtona¹.

První formální pojetí životního cyklu softwaru se objevuje v díle Winstona W. Royce *Managing the development of large software systems*², kde autor popisuje model, který byl později pány Bellem a Thayerem³ nazván „vodopádovým“, ačkoli on sám tento název nepoužil. Paradoxně byl tento model uveden jako příklad „nevhodného“ a „nefunkčního“ přístupu.

Ve své knize *The Mythical Man-Month*⁴ vydané v roce 1975 se Frederick P. Brooks věnuje mimo jiné i problematice prototypového vývoje softwaru. Vývoj v oblasti životních cyklů pak pokračoval definováním Inkrementálního modelu⁵ v roce 1985 a spirálového modelu vývoje B. Boehmem⁶ v roce 1986.

Vývoj v oblasti životního cyklu softwaru však zdaleka není ukončen. Z novějších dokumentů týkajících se této problematiky můžeme zmínit např. Manifest Agilního vývoje software⁷, jehož principy se v následujících letech staly stavebním pilířem mnoha agilních metod vývoje, Open Incremental Model⁸ publikovaný roku 2011, který autoři představují

¹ *Symposium on advanced programming methods for digital computers*. Washington D.C., USA, 1956.

² WINSTON, Royce. *Managing the development of large software systems*. [online]. 1970 [cit. 2014-03-31]. Dostupné z: <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>

³ Bell, T.E. a Thayer, T.A. *Software requirements: Are they really a problem?* [online]. 1976 [cit. 2014-03-31]. Dostupné z http://pdf.aminer.org/000/361/405/software_requirements_are_they_really_a_problem.pdf

⁴ BROOKS, F. *Mystical man-month - essays on software engineering*. Vyd. 1. Massachusetts: Addison-Wesley, 1982, 195 s. ISBN 02-010-0650-2.

⁵ DOD-STD-2167, MILITARY STANDARD: DEFENSE SYSTEM SOFTWARE DEVELOPMENT [online] 1985-06-04, [cit. 2014-03-31] Dostupné z: http://www.everyspec.com/DoD/DoD-STD/DOD-STD-2167_278/

⁶ BOEHM, B. *ACM SIGSOFT Software Engineering Notes* [online]. 1986-08-01, vol. 11, issue 4, s. 22-42 [cit. 2014-03-31]. DOI: 10.1145/12944.12948. Dostupné z: <http://portal.acm.org/citation.cfm?doid=12944.12948>

⁷ *Manifest agilního vývoje software*. [online]. [cit. 2014-03-31]. Dostupné z: <http://agilemanifesto.org/iso/cs/>

⁸ MANDAL, S., KANDAR, S. a RAY, P. Open Incremental Model- A Open Source Software Development Life Cycle Model (OSDLC). [online] International Journal of Computer Applications, 2011, vol. 21, no. 1 ProQuest Technology Collection. ISSN 09758887. Dostupné z: <http://dx.doi.org/10.5120/2473-3327>.

jako moderní alternativu použitelnou při vývoji open-source softwaru, či PARTH's model⁹ z roku 2010.

⁹ BIRENDRA, P. a KAUR, A. A Proposal for New Software Development Life Cycle Model: PARTH's Model. [online] Singapore: Research Publishing Services, 2010 ProQuest Technology Collection. Dostupné z: http://dx.doi.org/10.3850/978-981-08-7618-0_1352

2 Základní pojmy

Certifikace

Certifikace je v dnešní době nejčastěji chápána jako akt vystavení osvědčení (certifikátu) za účelem prokázání splnění požadavků stanovených pro tento certifikát. Tyto požadavky bývají stanoveny nejrozličnějšími certifikačními orgány, které jsou určitou garancí zachování kvality certifikace.

V normě ČSN EN ISO/IEC 17 021 je uvedeno „*Certifikace systému managementu, jakým je systém managementu kvality nebo systému environmentálního managementu organizace, je jedním z prostředků poskytování záruky, že organizace uplatňuje ve shodě se svou politikou pro odpovídající hlediska svých činností systém managementu.*“¹⁰. Z tohoto hlediska tak můžeme certifikaci chápat také jako proces ověřování, že daný objekt je v souladu s vytvořeným normalizovaným standardem.

Certifikační orgán

Certifikačním orgánem je chápána akreditovaná certifikační společnost, oprávněná provádět hodnocení shody s normou. Akreditace certifikačních orgánů zpravidla uděluje národní akreditační orgán daného státu. V České republice tuto funkci vykonává Český institut pro akreditaci, o.p.s.¹¹

Požadavky na certifikační orgán jsou dány různými mezinárodními normami v závislosti na tom, které oblasti certifikací se chce věnovat:

- ČSN EN ISO/IEC 17025:2005 pro zkušební laboratoře,
- ČSN EN ISO 15189:2007 pro zdravotnické laboratoře,
- ČSN EN ISO/IEC 17025:2005 pro kalibrační laboratoře,

¹⁰ ČSN EN ISO/IEC 17021:2011. *Posuzování shody: Požadavky na orgány poskytující služby auditů a certifikace systémů managementu*. 1. vyd. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011, s. 11.

¹¹ Český institut pro akreditaci, o.p.s.: O nás. *Český institut pro akreditaci, o.p.s.* [online]. [cit. 2013-12-13]. Dostupné z: <http://www.cia.cz/o-nas.aspx>

- ČSN EN ISO/IEC 17021:2011 pro certifikační orgány provádějící certifikaci systémů jakosti, systémů environmentálního managementu, systémů managementu bezpečnosti a ochrany zdraví při práci, systémů managementu bezpečnosti informací, systémů managementu bezpečnosti potravin a systému trvale udržitelného hospodaření v lesích,
- ČSN EN 45011:1998 / ČSN EN ISO/IEC 17065:2013 pro certifikační orgány certifikující produkty,
- ČSN ISO 14065:2008, nařízení Komise (EU) č.600/2012 pro ověřovatele emisí skleníkových plynů,
- ČSN EN ISO/IEC 17024:2003 / ČSN EN ISO/IEC 17024:2013 pro certifikační orgány provádějící certifikaci osob
- ČSN EN ISO/IEC 17020:2005 / ČSN EN ISO/IEC 17020:2012 pro inspekční orgány,
- ČSN EN ISO/IEC 17043:2011 pro poskytovatele zkoušení způsobilosti,
- nařízení ES č. 1221/2009 pro environmentální ověřovatele programů EMAS a dohled nad zahraničními environmentálními ověřovateli.¹²

Norma ISO/IEC 12207

Poslední přeloženou verzí této normy je Změna Amd. 1, která je českou verzí změny ISO/IEC 12207:1995/Amd. 1:2002. Tato norma je tak dnes již 18 let stará, resp. 11 let v případě její Změny, což je v oblasti informačních technologií téměř nepřekonatelnou překážkou pro její efektivní využívání. Pro potřeby této práce jsem se tedy rozhodl využívat aktuální anglickou verzi citované normy.

„The purpose of this International Standard is to provide a defined set of processes to facilitate communication among acquirers, suppliers and other stakeholders in the life

¹² Viz 11

*cycle of a software product.*¹³ – Cílem této mezinárodní normy je poskytnout definovaný soubor procesů k usnadnění komunikace mezi pořizovateli, dodavateli a dalších zúčastněných stran v životním cyklu produktu.

Z první kapitoly citovaného dokumentu pak jasně vyplývá, co je a není jejím předmětem. Norma samotná, navzdory svému názvu, nepojednává o konkrétním „doporučeném“ životním cyklu softwaru a jeho certifikaci. Zabývá se procesy s životním cyklem souvisejícími v celkovém kontextu organizace. Smyslem je poskytnout jasně definovanou terminologii a procesy, které by mohly být využívány v rámci co nejširšího okruhu situací a životních cyklů softwaru a umožnit tak efektivní komunikaci mezi jednotlivými zúčastněnými stranami.

Software

Software je v jednom z největších internetových slovníků definován jako „*written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory*“¹⁴ – *napsané programy, procedury nebo pravidla a příslušná dokumentace vztahující se k provozu počítačového systému a které jsou uloženy v paměti typu čtení/zápis*“.

Software je často rozdělován na dvě základní kategorie – systémový software a aplikační software.¹⁵

Systémový software

Úkolem systémového softwaru je zajistit základní nespecifické funkce počítače. Je zodpovědný za ovládání, integraci a kontrolu jednotlivých hardwarových komponent tak, aby jej uživatelé a ostatní programy vnímali a zejména mohli používat jako funkční celek

¹³ ISO/IEC 12207:2008. *Systems and software engineering: Software life cycle processes*. Second edition. United States of America: ISO/IEC-IEEE, 2008, s. 1.

¹⁴ *The Free Dictionary* [online]. 2012 [cit. 2013-12-16]. Dostupné z: <http://www.thefreedictionary.com/computer+software>

¹⁵ VARIA, Vishal. *ASSIGNMENT ON SOFTWARE, ITS TYPES, SYSTEM SOFTWARE, OPERATING SYSTEM, FUNCTION & ITS TYPES* [online]. Rajkot, 2011 [cit. 2013-12-16]. Dostupné z: http://www.academia.edu/4564920/ASSIGNMENT_ON_SOFTWARE_ITS_TYPES_SYSTEM_SOFTWARE_OPERATING_SYSTEM_FUNCTION_and_ITS_TYPES. Assignment. Department of Education, Saurashtra University.

bez nutnosti zabývat se nízko-úrovňovými operacemi jako je převod dat z paměti na disky či zobrazení textů.

Obecně se za systémový software považuje operační systém počítače a jeho základní programové součásti – správce souborů, hardwarové ovladače, nástroje pro správu a autentizaci uživatelů, síťové ovladače apod.

Aplikační software

Úkolem aplikačního softwaru je vykonávání specifických úkolů určitého typu, na který je specializovaný. Aplikační software může být jeden program s konkrétním využitím, např. prohlížeč obrázků, nebo také balík vzájemně propojených programů, např. Microsoft Office.

Open-source¹⁶

Open-source je v zásadě software, který může být volně užíván, měněn a sdílen. To však není vše. Jeho distribuční podmínky musí splňovat také následující kritéria:

Volná redistribuce – Licence nesmí žádným způsobem omezovat další šíření programu či jeho částí, včetně placeného prodeje.

Zdrojový kód – Distribuce softwaru musí zahrnovat i jeho zdrojový kód, ne jen zkompilovanou spustitelnou verzi. Pokud není vhodné zahrnovat zdrojový kód přímo k dané distribuci, musí mít jasně označen způsob získání příslušného zdrojového kódu například z webových stránek autora.

Odvozená díla – Licence musí umožňovat tvorbu odvozených děl a využívání částí zdrojového kódu v jiných projektech.

Žádná diskriminace – Licence nesmí diskriminovat žádnou osobu či skupinu osob, stejně tak jako využití ve specifických oblastech lidské činnosti.

Distribuce licence – Práva spojená s programem se musí vztahovat na všechny uživatele programu, bez ohledu na způsob jeho získání.

¹⁶ Open Source Initiative. *The Open Source Definition* [online]. [cit. 2014-04-14]. Dostupné z: <http://opensource.org/osd>

Licence nesmí být vázaná na specifický produkt – Práva spojená s programem musí být stejná pro větší celek, jehož je program součástí, stejně jako pro program samotný v případě, kdy je z tohoto celku vyseparován.

Licence nesmí omezovat jiný software – Například nesmí požadovat, aby veškerý další software s ní distribuovaný byl také open-source.

Licence musí být technologicky neutrální – Žádná ustanovení licence nesmí ukazovat na individuální technologii či styl uživatelského rozhraní.

Audit

*A planned and documented activity performed by qualified personnel to determine by investigation, examination, or evaluation of objective evidence, the adequacy and compliance with established procedures, or applicable documents, and the effectiveness of implementation.*¹⁷ – *Plánovaná a dokumentovaná aktivita prováděná kvalifikovanou osobou za účelem prokázání, zkoumáním, přezkoumáním a vyhodnocením objektivních důkazů přiměřenosti, přiměřenosti a shody se zavedenými postupy, použitelnými dokumenty a efektivitou implementace.*

Audity můžeme rozdělit v zásadě na dva typy: účetní audity a výkonnostní audity.

Cílem účetního auditu je ověřit účetnictví organizace z hlediska správnosti a souladu s právními požadavky daného státu. Ve většině vyspělých zemí je účetní audit povinný minimálně pro některé typy obchodních společností. V České republice mají povinnost auditu účetní uzávěrky akciové společnosti, které splnily minimálně jedno z následujících kritérií, a ostatní obchodní společnosti a družstva, které splnily aspoň dvě:

- celková hodnota aktiv společnosti převyšuje 40 milionů Kč,
- roční obrat převyšuje 80 milionů Kč,
- průměrný přepočtený stav zaměstnanců činí více než 50.

¹⁷ Audit. *Auditor Dictionary* [online]. [cit. 2014-04-14]. Dostupné z: <http://www.projectauditors.com/Dictionary2/1.8/index.php/term/62555c9cae53606f68555ca75a.xhtml>

Do skupiny výkonnostních auditů pak spadá široká škála převážně technických auditů, např. audity systémů řízení, audity projektů či energetické audity.

3 Životní cyklus softwaru

*SDLC is a process used by IT analysts in order to develop or redesign high quality software system which meets both the customer and the real world requirement taking into consideration all associated aspects of pros and cons of software testing, analysis and post process maintenance.*¹⁸ – Životní cyklus vývoje software je proces používaný IT analytiky k vývoji či předělání vysoce kvalitního softwarového systému, který by splňoval jak požadavky zákazníka tak reálného světa, zohledňující všechny výhody a nevýhody testování softwaru, analýzu a následnou údržbu.

Životní cyklus pokrývá všechny fáze vývoje softwaru, které na sebe plynule navazují. Pomocí vhodného životního cyklu lze popsat vývoj každého softwaru. Základními fázemi jsou analýza, návrh, vývoj, testování a finalizace.

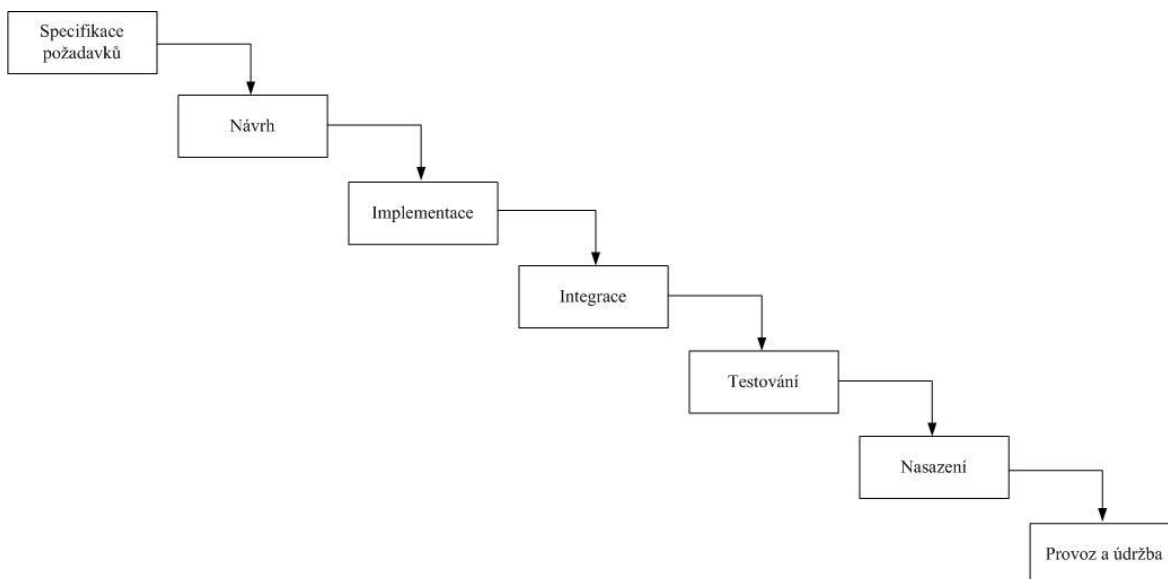
Níže se podíváme na vybrané modely životních cyklů softwaru.

3.1 Vodopádový model

Vodopádový model představuje vývoj jako lineární sekvenci jednotlivých kroků. Linearita je reprezentována faktem, že žádný krok nemůže začít dříve, než je plně ukončen krok předchozí. Z výše uvedeného vyplývá, že v tomto modelu nejsou definovány procesy určené k návratu do předchozího bodu vývoje. Různé projekty využívající tento model tak mají pro tyto případy nastaveny různé postupy.

Tento model je ze všech zde uvedený úplně nejstarším. Ve své době plně vyhovoval požadavkům, které byly na vývoj softwaru kladeny, neboť jejich prostředí bylo poměrně stabilní a nebyla tak potřeba orientovat se na změny požadavků.

¹⁸ What is SDLC. *SDLC Tutorials* [online]. [cit. 2014-01-02]. Dostupné z: <http://www.sdlic.ws/what-is-sdlic/>



Obrázek č. 1 – Vodopádový model

Zdroj: vlastní zpracování.

Systemové požadavky

Specifikace požadavků na software je prvním krokem. Tato fáze začíná ve chvíli, kdy dojde k objevení problému, který má navrhovaný software řešit. Účelem této fáze je identifikovat cíle, kterých musí být dosaženo, odhadnout výhody nového systému před systémem stávajícím a také prozkoumat přidružené oblasti, které tvořené řešení ovlivní.

Součástí je také vytvoření obchodního projektu pro vytvářený software – vyjádření očekávaných nákladů, přínosů a prostředků, které budou pro jeho realizaci zapotřebí. Obchodní projekt musí managementu poskytnout podklady ke kvalifikovanému rozhodnutí o jeho případné realizaci před tím, než dojde k vyčlenění podnikových zdrojů.

Návrh

Návrh představuje řešení v hladině softwarové a hardwarové architektury. Jeho cílem je identifikovat jednotlivé části potřebné k uspokojení požadavků stanovených v předchozím kroku. Řeší se také požadavky na výkon softwaru a jeho zabezpečení proti neoprávněné manipulaci a zneužití, tvorba datových skladů a omezení, vhodné vývojové prostředí a programovací jazyk. Je vhodné alespoň naznačit možná problémová místa a návrhy řešení těchto problémů.

Výstupem této fáze by měl být jeden či více návrhů, které budou vzájemně porovnány z hlediska jejich optimálnosti a následně využity v další fázi vývoje.

Implementace

Tato fáze spočívá ve faktickém sestavení jednotlivých bloků softwaru tak, jak byly specifikovány v předchozím kroku. Programátoři a další potřební specialisté (odborníci přes user interface, databázové systémy aj.) postupně tvoří jednotlivé úseky a odstraňují chyby tak, aby svou část připravili na další fázi.

Integrace

V této fázi dochází k vzájemnému propojení jednotlivých součástí vytvořených v předchozím kroku. Vytvoří se propojovací prostředky k umožnění jejich vzájemné komunikace. Dojde k základnímu otestování vytvořeného celku. Posledním krokem této fáze je příprava testovacích dat.

Testování

V tomto kroku dochází k testování softwaru jako celku s částí reálných dat. Cílem je ověřit, že ve vytvořeném softwaru neexistují žádné chyby a že jako celek plně pokrývá potřeby požadavky stanovené v první fázi.

Fáze je prakticky rozdělena na tři části – testování jednotlivých modulů, testování celku a formální test u zákazníka pro ilustraci schopnosti vytvořeného programu splnit svůj účel a uspokojit zákaznickovy potřeby. V případě nalezení defektů jsou tyto zaregistrovány a předány týmu, který bude zajišťovat nasazení softwaru.

Během této fáze by měla být také vytvořena systémová dokumentace, uživatelské příručky a další potřebná dokumentace.

Nasazení

Nasazení se realizuje ve chvíli, kdy byly dokončeny veškeré naplánované testy implementovaného softwaru a tento byl schválen jako vhodný k použití.

Tato fáze zahrnuje jak přípravu samotného produktu na použití, tak přípravu okolního prostředí v organizaci, kde bude software nasazen, např. konfiguraci dalších aplikací, které s ním budou spolupracovat, či nastavení databázových serverů pro čtení a zápis potřebných dat.

Samotné předání softwaru bylo dříve prováděno prostřednictvím fyzického média, v dnešní době je však možné toto obstarat prostřednictvím sítě internet. Dodávaný software je obvykle označen formálním revizním číslem, které zajišťuje kontrolu, zda byla opravdu dodána jeho finální verze.

Provoz a údržba

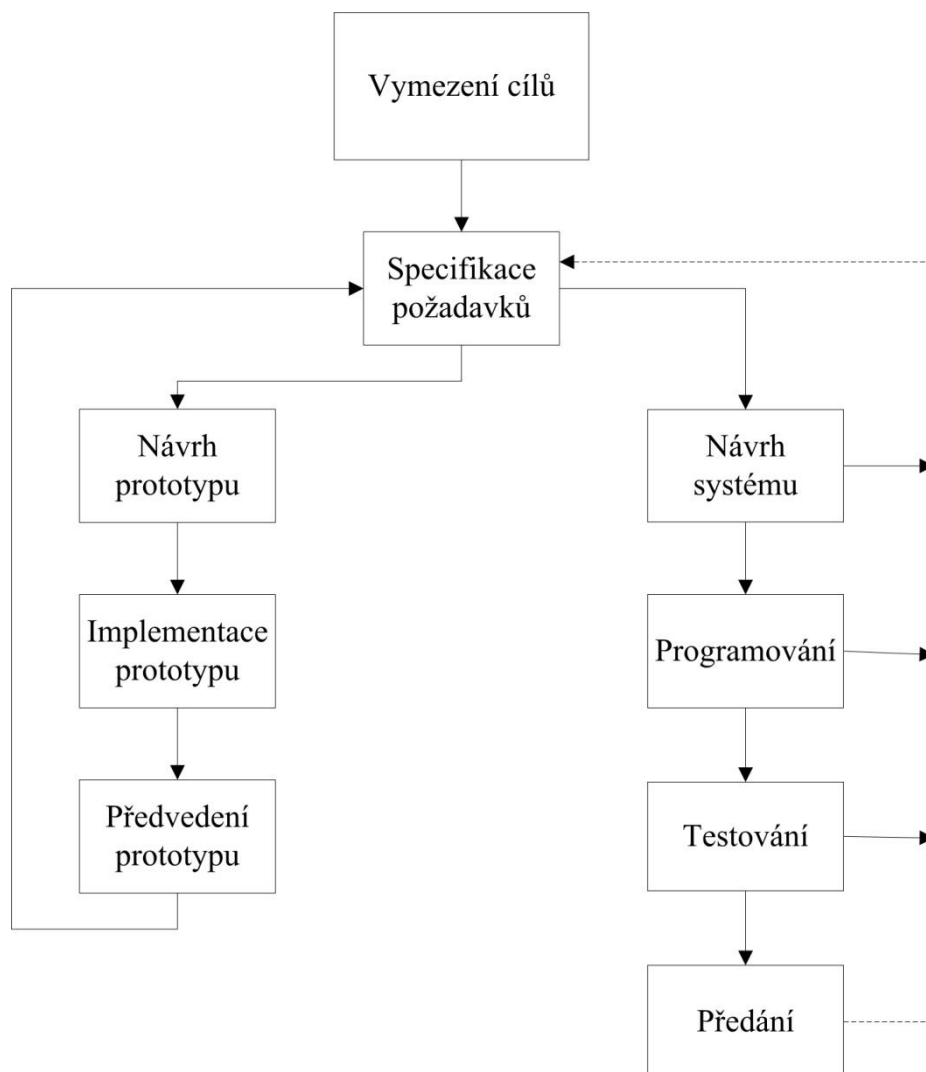
Poslední fáze tohoto životního cyklu spočívá v udržování softwaru v provozu. Zahrnuje také drobné modifikace s cílem zlepšení výkonu aplikace či odstranění chyb a nedostatků. Každá změna by měla být zaznamenána a odpovídajícím způsobem by měly být změněny i systémová dokumentace a další dokumenty.

3.2 Prototypový model

Prototypový přístup (známý též jako evoluční přístup) byl reakcí na problémy zjištěné u finálních verzí aplikací vytvořených prostřednictvím vodopádového modelu. Tyto problémy obvykle vznikají v důsledku změn v požadavcích během vývoje softwaru nebo v důsledku špatného pochopení potřeb zákazníka vývojovým týmem. Tyto nedostatky často vedly k potřebě kompletního přepracování vytvořené aplikace, což časem vyústilo v tvorbu prototypů.

Jako prototyp se používá zpravidla částečná implementace celého systému či úplná aplikace některých jeho částí. Tato se provádí v co nejkratším čase (úspora nákladů) a v maximální možné funkčnosti tak, aby získal zákazník představu o možnostech finálního produktu. Na základě připomínek od zákazníka jsou pak požadavky na systém upravovány a zpřesňovány tak, aby se další prototyp přiblížil jejich splnění. Ve chvíli, kdy je zákazník plně spokojen s představeným prototypem, dochází k tvorbě finální verze softwaru.

Tento model se obtížně praktikuje u velkých projektů z důvodu nutnosti tvorby značného množství vytvořených prototypů.



Obrázek č. 2 – Prototypový model

Zdroj: vlastní zpracování.

3.3 Spirálový model

Spirálový model je z určitého pohledu kombinací obou předchozích. Spojuje v sobě jak iterační prvky prototypového modelu, tak sekvenční přístup modelu vodopádového. Ze svojí povahy je velice vhodným pro použití u softwaru, který ve více či méně pravidelných intervalech vychází v nových verzích (typicky open-source projekty).

Princip tohoto modelu spočívá v řetězení fází lineárního vodopádového přístupu do de facto nekonečného cyklu, kdy výstupy z dokončeného cyklu figuruji jako vstupy v cyklu následujícím.

Fáze spirálového modelu:¹⁹

Komunikace se zákazníkem

Průběžná komunikace se zákazníkem o jeho potřebách a požadavcích. Cílem je komplexní porozumění systémovým požadavkům, které jsou na software kladeny.

Plánování

Vytvoření harmonogramu, sestavení rozpočtu a vyčlenění prostředků potřebných ke zhotovení.

Riziková analýza

Identifikace, monitorování a odhadování výše technických i řídicích rizik spojených s plánovaným projektem.

Návrh

Sběr požadavků a návrh samotného systému.

Tvorba a nasazení

Kódování, testování a finální nasazení hotového softwaru u konečného zákazníka, vytvoření potřebné softwarové dokumentace, uživatelského manuálu apod.

Zpětná vazba zákazníka

Vyhodnocení softwaru zákazníkem. Výstupy z hodnocení jsou převzaty jako vstupy do dalšího cyklu vývoje.

¹⁹ Spiral model. *SDLC Tutorials* [online]. [cit. 2014-02-10]. Dostupné z: <http://www.sdgc.ws/spiral-model/>

3.4 Inkrementální model

Tak jako je spirálový model faktickým sériovým zapojením několika vodopádových modelů za sebou do komplexního celku, tak je inkrementální model zapojením paralelním. Podstatou tohoto modelu je „rozbití“ softwaru na dílčí, samostatně funkční celky, které jsou postupně vyvíjeny a připojovány k již fungujícím částem.

Lze tak využít toho, že jsou jednotlivé, již funkční, části průběžně nasazovány v koncovém prostředí u zákazníka. Dochází tak k eliminaci dlouhých dodacích lhůt běžných u jiných vývojových modelů i vysokých jednorázových nákladů na pořízení systému. V případě inkrementálního modelu lze totiž fakturovat i po jednotlivých částech systému, které již byly zprovozněny.

3.5 RAD

RAD je zkratka pro rapid application development, česky něco jako „rychlý vývoj aplikací“. Jeho podstatou je minimalizace plánování a rychlé vytváření funkčních prototypů. Díky tomu je možné tvořit aplikace daleko rychleji a také jednoduše měnit požadavky. Omezení plánování na nezbytné minimum pak nahrazuje intenzivní komunikace s konečnými uživateli softwaru, kteří jsou aktivně zapojeni do vývoje formou připomínkových řízení či různých konzultací. Metoda RAD sestává ze čtyř fází:

Plánování požadavků

Uživatelé, management a tvůrci diskutují nad požadavky, které bude muset tvořený software uspokojovat. Výstupem by měl být obecný konsenzus ohledně toho, které požadavky budou uspokojeny, a které budou odloženy jako nerelevantní.

Návrh uživatelského designu

Během této fáze analytici, za intenzivní spolupráce s uživateli jednotlivých komponent, navrhnu architekturu systému, vstupy a výstupy.

Konstrukce

Samotná tvorba softwaru programátory, ověření schopnosti softwaru uspokojovat stanovené požadavky a testování funkčnosti.

Dokončení

Poslední fáze zahrnuje nasazení softwaru u zákazníka, finální testy, konfiguraci softwaru a školení uživatelů.

3.6 Agilní přístupy

Výše zmiňované přístupy považují tvorbu softwaru za pevně definovaný proces, který je možné jednoznačně popsat. Agilní přístupy naopak tvrdí, že proces tvorby je empirický, nemá cenu jej popisovat, pouze monitorovat a přizpůsobovat realitě. Agilní metodiky tak nepopisují procesy při vývoji softwaru, definují pouze určité principy a praktiky.

Existuje značné množství různých agilních metod, ale v zásadě všechny vycházejí ze zásad definovaných v Manifestu agilního vývoje softwaru.²⁰

- Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného softwaru.
- Vítáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
- Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.
- Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.
- Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.

²⁰ *Principy stojící za Agilním Manifestem.* [online]. [cit. 2014-04-15]. Dostupné z: <http://agilemanifesto.org/iso/cs/principles.html>

- Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.
- Hlavním měřítkem pokroku je fungující software.
- Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.
- Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.
- Jednoduchost--umění maximalizovat množství nevykonané práce--je klíčová.
- Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.
- Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.

Mezi nejrozšířenější agilní metodiky patří např. Scrum či Extrémní programování.

4 Norma ISO/IEC 12207:2008

Norma ISO/IEC 12207 (dále jen norma) byla vydána v první verzi v roce 1995. Byla prvním pokusem o vytvoření mezinárodního standardu, který by pojmenovával procesy spojené s životním cyklem softwaru a umožňoval tak všem zapojeným stranám efektivně komunikovat.

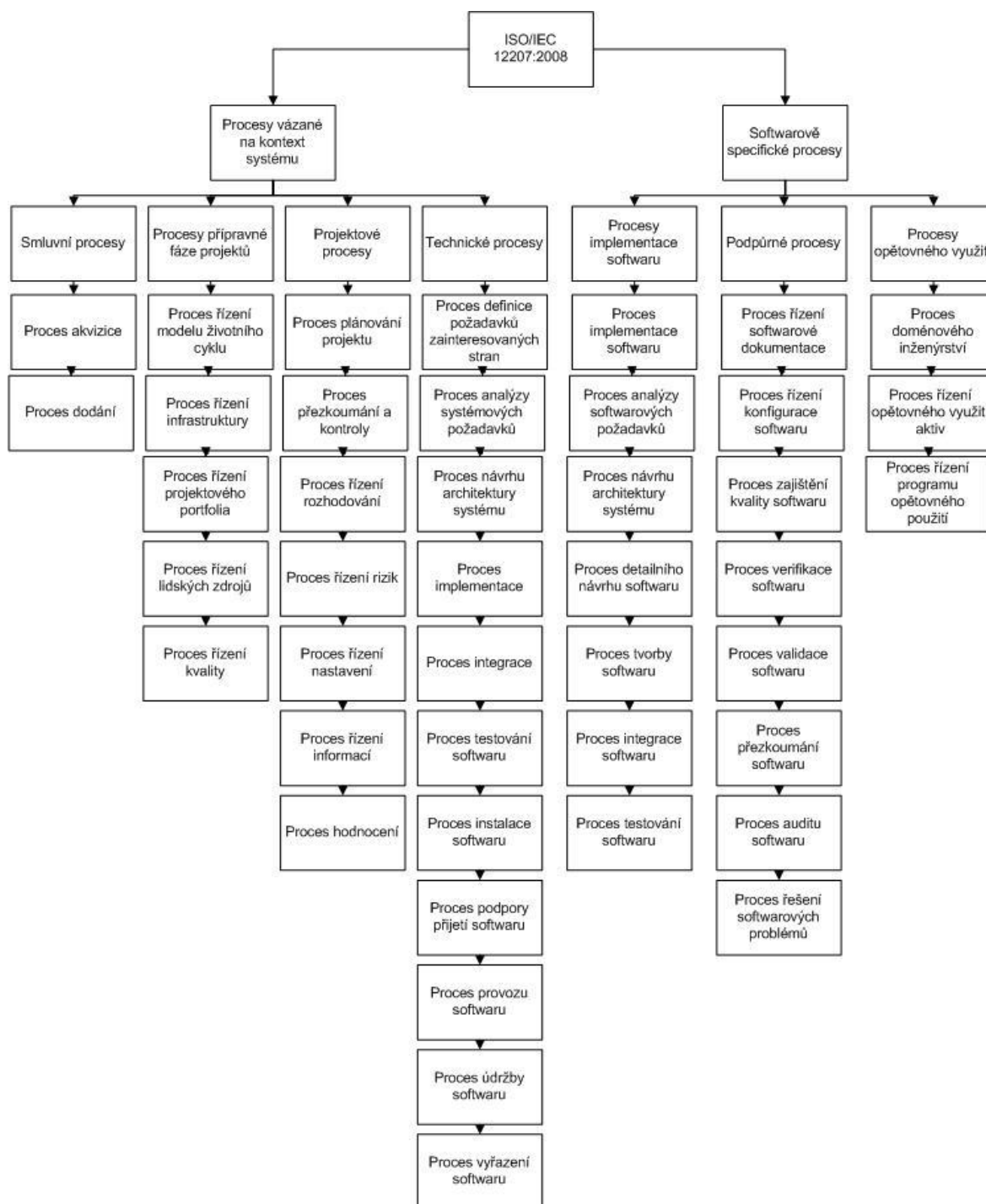
„Tato mezinárodní norma není určena pro předepisování názvu, formátu nebo explicitního obsahu vytvářené dokumentace.“²¹

Tato norma nepředepisuje žádný specifický model životního cyklu či metodu vývoje softwaru. Shoda s normou je definována jako provedení všech procesů, činností a úloh z ní vybraných v procesu přizpůsobení. Tento proces je podrobně popsán v příloze A výše zmíněné normy. V případě, že se zainteresované strany rozhodnou tuto normu použít, tak musí procesy, činnosti a úlohy v ní uvedené aplikovat na zvolený model životního cyklu softwaru tak, aby byly zajištěny všechny požadavky figurující mezi výstupy z procesu přizpůsobení.

Celá norma je zaměřena zejména na popis procesů souvisejících s kompletním životním cyklem softwaru od úvodních dohod mezi pořizujícím a dodavatelem až po konečnou implementaci a údržbu. Je sestavena jako softwarově specifický ekvivalent normy ISO/IEC 15288, která se zabývá životními cykly systému jako celku a neřeší specifika dílčích softwarových částí.

Z tohoto důvodu byla ve verzi z roku 2008 také sjednocena struktura těchto dvou norem. Obě jsou tak zaměřeny na popis procesů, přičemž jsou sjednoceny i jejich názvy a terminologie.

²¹ ČSN ISO/IEC 12207. *Informační technologie: Procesy v životním cyklu softwaru*. Praha: ČESKÝ NORMALIZAČNÍ INSTITUT, 1997, str. 5



Obrázek č. 3 – Rozdělení procesů normy ISO/IEC 12207:2008

Zdroj: vlastní zpracování

Poněkud paradoxně lze do budoucna očekávat spíše negativní vývoj zájmu o certifikaci dle této normy. Důvodem je stále rostoucí obliba agilních metodik vývoje mezi programátory a potažmo jejich adaptace do fungování softwarových firem. Zatímco agilní metodiky obecně kladou důraz spíše na neformální vývoj bez zbytečného zatěžování se formalitami,

vyžaduje norma ISO/IEC 12207 přesný opak. K získání certifikace je nutné mít ve firmě přesně nastavené značné množství procesů, tyto popsány, průběžně analyzovány a zlepšovány. Je také potřeba vést administrativu celého systému a uchovávat záznamy o provedení jednotlivých činností. Vše ve faktickém rozporu s Manifestem agilního vývoje.

Dá se tak do budoucna očekávat, že zájem o certifikaci dle normy tak zůstane vázán na několik poměrně specifických oblastí. Jedná se o oblasti, kde nelze tvorbu systémů realizovat agilním vývojem z důvodu nutnosti ošetření všech formalit. Mezi tyto oblasti patří například letectví, ať už se bavíme o softwarovém vybavení pozemních stanovišť nebo integrovaném softwaru v letadlech, navigační systémy vesmírných družic a plavidel či zdravotnické prostředky. Ve všech těchto oblastech existuje potřeba co nejformálnějšího vývoje s průběžnou kontrolou a možností zpětně dohledat zodpovědné osoby. Jedná se totiž o oblasti, kde každé softwarové selhání může být katastrofální.

5 Praktická část

V době mého nástupu na roční řízenou praxi začala společnost plánovat nabídku auditů životního cyklu softwaru dle normy ISO/IEC 12207:2008. S ohledem na nedostatečné předchozí zkušenosti s prováděním auditů dle této normy vznikla potřeba vytvoření nějakého manuálu, metodického postupu či jiné pomůcky pro auditory. Tohoto úkolu jsem se nakonec ujal a rozhodl jsem se jej řešit jako téma své bakalářské práce.

Na následujících stránkách se tak budu věnovat řešení svěřeného problému.

5.1 Vstupní nástroje auditora

Při provádění auditu má auditor k dispozici několik pomůcek, jejich úkolem je auditování v maximální možné míře zjednodušit. Na následujících stránkách se na tyto nástroje blíže podívám a rozeberu, proč jsem se rozhodl či nerozhodl pro jejich zpracování.

5.1.1 Zkušenosti auditora

Ač se nejedná o nástroj v klasickém slova smyslu, jsou zkušenosti auditora bezpochyby jeho největším pomocníkem. S každým provedeným auditem získává další a další znalosti možných řešení splňujících požadavky normy. Získává tak v dlouhodobém horizontu určitý rozhled, který mu umožňuje lépe posoudit vhodnost opatření zavedených v auditované organizaci.

Samozřejmě i tato pomůcka by šla jistým způsobem zformalizovat. Jednotliví auditori si dlouhodobě se svými kolegy navzájem vyměňují své zkušenosti s možnostmi naplnění konkrétních požadavků norem v rámci různých porad a setkání. Cesta formalizace by tak v tomto případě spočívala ve vytvoření určitého „seznamu možných řešení“. Takový seznam by mohla tvořit například tabulka, jejímiž sloupci by byly jednotlivé požadavky normy a možná řešení, se kterými se auditori v minulosti setkali.

Číslo článku	Název požadavku	Možnosti plnění
4.1	Všeobecné požadavky	Analýza podnikových procesů potřebných pro systém managementu kvality; Nastavené metriky pro jejich monitoring
4.2	Požadavky na dokumentaci	
4.2.1	Obecně	Příručka jakosti; Prohlášení o politice a cílech kvality

Obrázek č. 4 – Ukázka návrhu dokumentu založená na ČSN EN ISO 9001:2008

Zdroj: vlastní zpracování

V současné době není tento nástroj ve společnosti vůbec využíván. Jelikož se de facto jedná o pouhé formalizování již existující pomůcky a její sdílení mezi kolegy, doporučoval bych její vytvoření pro všechny stávající auditované oblasti. Vytvořené seznamy by tak mohly sloužit k rychlejšímu zaučení zejména méně zkušených auditorů.

Pro potřeby vytvoření postupu pro provádění auditů dle normy ISO/IEC 12207:2008 však tento nástroj není vhodný, jelikož praktické zkušenosti auditorů s touto normou jsou téměř nulové.

5.1.2 Plán auditu

Sestavení plánu auditu je povinnou součástí přípravy na audit. Při jeho tvorbě je brána v potaz jak určená časová náročnost prováděného auditu, tak jeho obsahová stránka. První částí přípravy plánu auditu je stanovení celkové délky jeho trvání. Ta je stanovena matematickým vzorcem, který zohledňuje zejména velikost auditované oblasti, předmět auditu a celkovou komplexnost systému.

Druhou částí přípravy plánu auditu je pak samotné rozdělení času na jednotlivé úseky. Toto je již čistě subjektivní záležitost auditora, který při alokaci potřebného času vychází zejména ze svých zkušeností a, v případě, kdy v organizaci neprovádí audit poprvé, výsledků předchozích auditů. Například tak vyhradí větší časový úsek na kontrolu oblasti, ve které byly při minulé kontrole nalezeny nedostatky.

Plán auditu je tak seznamem jednotlivých oblastí, kterým se auditor musí věnovat (ke kterým potřebuje získat důkaz o splnění konkrétních požadavků normy) a časového

harmonogramu. Při samotné realizaci auditu pak slouží auditorovi jako vodítko, které oblasti by se měl věnovat do větší hloubky a kterým směrem by měl audit postupovat.

Plán ... auditu pro posouzení shody systému ... podle ČSN EN ISO 9001:2009 konaného dne ...

Organizace:

Vedoucí auditor:
Auditor:

|

***Nehodící se**

škrtněte

Probíhá od - do	Článek normy ČSN EN ISO 9001:2009	Předmět auditu/auditovaný proces dle rozsahu na certifikátu (dle žádosti o certifikaci)	Místo auditu (ústředí/lokalita/dočasná pracoviště)	Auditor	Kontaktní osoba
1. den					

Obrázek č. 5 – Ukázka plánu auditu

zdroj: vnitřní dokument EZÚ, s.p.

Jak již bylo zmíněno, sestavení konkrétního plánu auditu je povinným krokem při přípravě na audit. Tento dokument se tak s každým prováděným auditem mění a není tak možné jej vytvořit jako univerzální pomůcku. Z tohoto důvodu není jeho tvorba vhodným řešením mého problému.

5.1.3 Check list

Check listy jsou další důležitou pomůckou auditora. Vytváří si je obvykle sama certifikační organizace prostřednictvím svých zaměstnanců a to pro každý mezinárodní standard, podle něžž se certifikace provádí. Vytvářejí se obvykle před provedením prvního auditu dle dané normy a podléhají pravidelné aktualizaci – vždy s novou verzí příslušné normy.

Jedná se v podstatě o tabulku obsahující zpravidla následující informace:

- číslo požadavku a jeho název – informace převzaté z normy,

- rozlišení, čeho se požadavek týká – např. vyžaduje vytvoření vnitřní směrnice/politiky, vytváření záznamů o provedení určité činnosti či nastavení vnitřního procesu,
- vysvětlení požadavku – krátký článek popisující, co bylo konkrétním požadavkem míněno, co má auditor v organizaci ověřovat; některé normy obsahují u jednotlivých článků rovnou i vysvětlující popis, u ostatních musí tvůrce check listu vzít v potaz vlastní zkušenosti, nápady a znalosti z případných školení.

Jedná se o nejobecnější pomůcku, která nepředpokládá rozsáhlé praktické znalosti na straně auditora. Ty by naopak mohly být při její tvorbě na škodu, neboť auditor s rozsáhlými zkušenostmi trpí určitou formou profesní slepoty a nemusí si tak uvědomit některé možnosti, které napadnou nezaujatého pozorovatele (tzv. „out-of-the-box thinking“).

Z výše uvedených důvodů jsem se rozhodl vytvořit pro řešení mého problému právě check list.

5.2 Rozdělení procesů

Prvním krokem přípravy nového check listu bylo celkové seznámení se s normou. Musel jsem vytvořit české názvosloví jednotlivých procesů a jejich skupin a pochopit jejich význam a výstupy.

Samotná norma rozděluje procesy v ní obsažené na dvě velké skupiny – Procesy vázané na kontext systému (body 5.2.1 až 5.2.4) a Softwarově specifické procesy (5.2.5 až 5.2.7).

První zmíněná kategorie jsou procesy, které fakticky ještě nesouvisí se samotným softwarem. Jedná se o procesy systémové vyplývající z kontextu organizace. Tvoří určitou „přípravnou fázi“, kterou organizace musí zrealizovat před samotnou tvorbou softwaru.

Druhou kategorií jsou pak procesy vázané přímo na daný softwarový produkt. Jedná se zejména o procesy související s implementací softwaru a jeho následným provozem a dalším využíváním.

5.2.1 Smluvní procesy

Tyto procesy definují činnosti nutné k uzavření dohody mezi dvěma organizacemi. Všechny tyto procesy vycházejí zejména z moderních konvencí v oblasti řízení a právních požadavků na uzavírání smluv apod.

Do této kategorie spadají dva hlavní procesy: akvizice; dodání.

5.2.2 Procesy přípravné fáze projektů

Tyto procesy ovlivňují schopnost organizace získat a poskytovat produkt či službu skrz iniciaci, podporu a řízení projektů. Zajišťují projektům nezbytné zdroje a infrastrukturu tak, aby tyto mohly úspěšně naplňovat cíle organizace.

Do této kategorie spadá pět následujících procesů: řízení modelu životního cyklu; řízení infrastruktury; řízení projektového portfolia; řízení lidských zdrojů; řízení kvality.

5.2.3 Projektové procesy

Pro potřeby tohoto mezinárodního standardu byl projekt vybrán jako kontext pro popsání procesů souvisejících s plánováním, vyhodnocováním a řízením. Projektové procesy jsou rozděleny na dvě hlavní kategorie.

Proces plánování projektu

Smyslem procesu je vytvořit a šířit efektivní a reálně uchopitelné projektové plány. Proces určuje cíle projektu, jednotlivé aktivity, úkoly a zdroje.

Procesy přezkoumání a kontroly projektu

Smyslem těchto procesů je zabezpečit komplexní řízení projektu ve všech jeho oblastech. Tato skupina zahrnuje následující dílčí procesy: řízení rizik; řízení nastavení; řízení informací; hodnocení.

5.2.4 Technické procesy

Úkolem technických procesů je definovat požadavky na systém, přeměnit požadavky na efektivní produkt, rozšířit produkt, používat jej a udržovat jej v provozu až do jeho vyřazení z provozu.

Tato skupina sestává z následujících dílčích procesů: definice požadavků zainteresovaných stran; analýzy systémových požadavků; návrhu architektury systému; implementace; integrace; testování softwaru; instalace softwaru; podpory přijetí softwaru; provozu softwaru; údržby softwaru a vyřazení softwaru.

5.2.5 Procesy implementace softwaru

Cílem těchto procesů je vytvořit specifický systémový element zabudovaný do softwaru. Tyto procesy transformují specifické chování, interface a implementační omezení do činností ústících ve vytvoření systémového elementu, který plně uspokojuje systémové požadavky.

Do této skupiny spadají následující procesy: implementace softwaru; analýzy softwarových požadavků; návrhu softwarové architektury; detailního návrhu softwaru; tvorby softwaru; integrace softwaru; testování softwaru.

5.2.6 Podpůrné procesy

Tato skupina obsahuje procesy řešící specifické problémy vázané na software. Jsou pevně provázány s procesy implementace softwaru, ale mají odlišný smysl – přispívají k úspěchu a kvalitě softwarového projektu.

Do této skupiny spadá následujících osm procesů: řízení softwarové dokumentace; řízení konfigurace softwaru; zajištění kvality softwaru; verifikace softwaru; validace softwaru; přezkoumání softwaru; auditu softwaru; řešení softwarových problémů.

5.2.7 Procesy opětovného využití

Do této skupiny patří tři procesy zaměřené na opakované využívání již vytvořeného softwaru v organizaci mimo rámec jeho původního zaměření. Jedná se o následující procesy: doménového inženýrství; řízení opětovného použití aktiv; řízení programu opětovného použití.

5.3 Vytvoření struktury

Pro potřeby mnou vytvářeného check listu jsem se nakonec rozhodl využít standardní formu tabulky. První dva sloupce jsou z pochopitelných důvodů tvořeny číslem jednotlivých článků normy a jejich názvy.

Jelikož je celá norma procesně orientovaná, je její struktura poněkud složitější, než bývá zvykem u jiných norem. Každý proces má několik podkapitol – význam, výstupy a jednotlivé aktivity, které je třeba v rámci procesu implementovat, provádět a sledovat. Jelikož norma pojmenovává celkem 43 procesů a každý z nich má v průměru 4 aktivity, bylo by nereálné v rámci auditu kontrolovat provádění každé z nich. Navíc i pro samotnou organizaci by uchovávání záznamů o každé dílčí aktivitě představovalo značnou administrativní zátěž.

Z výše uvedených důvodů jsem se tak rozhodl věnovat ve vytvářeném check listu „zaměření auditu“, které vystihuje povinné výstupy jednotlivých procesů. Auditor tak bude ověřovat přímo výsledky podnikových činností. Na základě věcné souvislosti pak lze rozumně předpokládat, že v situaci, kdy existují povinné výstupy v náležité kvalitě a se všemi požadovanými formalitami, byly správně vykonány i činnosti, které k jejich vytvoření vedly.

Posledním sloupcem check listu pak jsou „možná zjištění“, kde navrhuji některé možnosti, jakým způsobem lze doložit daná zjištění z auditu. V tomto sloupci jsem vycházel zejména z vlastních nápadů na ověření a také z nápadů odborného konzultanta mojí bakalářské práce.

Číslo článku	Název článku	Zaměření auditu	Možná zjištění
--------------	--------------	-----------------	----------------

Obrázek č. 6 – Záhloví check listu

Zdroj: vlastní zpracování

5.4 Analýza výstupů procesů

Předposledním krokem ve zpracování check listu byla analýza požadovaných výstupů jednotlivých procesů formulovaná do „zaměření auditu“ a „možná zjištění“.

Proces akvizice

Zaměření auditu: Potřeby a cíle akvizice. Cíle a kritéria přijetí akvizice. Dohoda/smlouva jasně stanovující postavení, očekávání a odpovědnosti obou stran. Vybrán dodavatel a dodán produkt uspokojující potřeby. Byly dodrženy stanovené náklady/rozpočet atd.

Možná zjištění: V průběhu jednání se zákazníkem vzniká technická specifikace zakázky, která je součástí finální nabídky.

Proces dodání

Zaměření auditu: Identifikován nabyvatel produktu. Odpověď na požadavek/poptávku nabyvatele. Dohoda o dodání produktu. Produkt uspokojuje schválené požadavky. Produkt je dodán a nainstalován dle schválených požadavků.

Možná zjištění: Proces dodání je jasně vymezen v uzavírané smlouvě.

Proces řízení modelu životního cyklu

Zaměření auditu: Politiky a postupy pro řízení životního cyklu SW. Definovány odpovědnosti a pravomoci. Procesy, modely a postupy životního cyklu jsou definovány, udržovány a zlepšovány.

Možná zjištění: Průběh procesů životního cyklu návrhu, implementace a udržování dodávaného SW produktu je naplánován a odsouhlasen podle uznávaných standardů v rámci smluv s dodavateli.

Proces řízení infrastruktury

Zaměření auditu: Požadavky na podporu procesů infrastrukturou. Definovány a specifikovány elementy infrastruktury. Tyto elementy jsou zajištěny a implementovány. Je udržována a zlepšována stabilní a spolehlivá infrastruktura.

Možná zjištění: Hardwarová i softwarová infrastruktura je jasně definována. Seznam používaných aplikací.

Proces řízení projektového portfolia

Zaměření auditu: Obchodní příležitosti, investice a potřeby. Zdroje pro každý projekt. Zodpovědnost za projekty stanovena. Projekty splňující kritéria udržovány. Projekty nesplňující kritéria ukončeny.

Možná zjištění: Management projektů probíhá prostřednictvím plánování, dokladování a přezkoumávání pomocí některé z aplikací pro projektové řízení.

Proces řízení lidských zdrojů

Zaměření auditu: Identifikovány dovednosti vyžadované projektem. Zajištěny lidské zdroje pro projekty. Dovednosti zaměstnanců udržovány a rozvíjeny. Řešeny konflikty lidských zdrojů mezi projekty. Individuální znalosti a dovednosti jsou shromažďovány a využívány napříč organizací.

Možná zjištění: Jsou identifikovány požadované dovednosti pro funkce a dílčí pozice. Záznamy o alokaci lidských zdrojů k jednotlivým projektům.

Proces řízení kvality

Zaměření auditu: Politiky a postupy řízení kvality. Cíle řízení kvality. Zodpovědnost za řízení kvality. Spokojenost zákazníků hodnocena. Opatření v případě, že cíle kvality nejsou plněny.

Možná zjištění: Certifikace dle ISO 9001. Příručka jakosti. Podnikové cíle v oblasti jakosti.

Proces plánování projektu

Zaměření auditu: Stanoven rozsah projektu. Vyhodnocena schopnost dosažení cílů. Úkoly a zdroje odhadnuty. Interface pro vnitřní i vnější komunikaci. Plán realizace projektu.

Možná zjištění: Management projektů probíhá prostřednictvím plánování, dokladování a přezkoumávání pomocí některé z aplikací pro projektové řízení.

Proces přezkoumání a kontroly

Zaměření auditu: Postup projektu monitorován. Interface pro komunikaci monitorován. Podniknuty kroky pro odstranění odchylek od plánu. Dosahovány a zaznamenávány cíle projektu.

Možná zjištění: Záznamy z porad, přezkoumání projektů managementem.

Proces řízení rozhodování

Zaměření auditu: Strategie řízení rozhodování. Definovány alternativní možnosti řešení. Vybrána nejvhodnější alternativa. Dokumentace o rozhodnutí a argumentech pro ně.

Proces řízení rizik

Zaměření auditu: Stanoven rozsah řízení rizik. Přiměřené strategie řízení rizik. Průběžná identifikace a analýza rizik. Nastaveny metriky pro hodnocení rizik. Přijata vhodná opatření k odstranění rizik.

Možná zjištění: Rizikové analýzy projektů, SWOT analýzy.

Proces řízení nastavení

Zaměření auditu: Strategie řízení nastavení. Definovány položky vyžadující řízení nastavení. Pravidla (zásady) nastavení. Monitoring změn předmětů podléhajících řízení nastavení. Kontrola nastavení uvolněných předmětů.

Možná zjištění: Procesní příručka, manuály pro konfiguraci.

Proces řízení informací

Zaměření auditu: Identifikovány informace k řízení. Forma interpretace informací. Požadavky na převod a odstranění informací. Záznamy o stavu informací. Informace musí být aktuální, úplné a správné. Informace zpřístupněny správným osobám.

Možná zjištění: Zavedená aplikace pro sdílení informací, aplikace pro řízení projektů se sdílenými přístupy, firemní intranet.

Proces hodnocení

Zaměření auditu: Identifikovány informační potřeby řídicích procesů. Vhodná sada metrik pro hodnocení. Plánování hodnocení. Požadovaná data posbírána, analyzována a uchována. Hodnotící procesy jsou evaluovány. Zlepšení jsou komunikována k majiteli hodnotícího procesu.

Možná zjištění: Záznamy o hodnocení. Nastavené hodnotící procesy a oblasti v rámci používané aplikace pro projektové řízení.

Proces definice požadavků zainteresovaných stran

Zaměření auditu: Určen kontext a charakteristiky použití služby. Omezení systémového řešení. Požadavky dohledatelné. Zásady definování systémových požadavků. Zásady pro ověření shody. Zásady pro vyjednávání o dodání.

Možná zjištění: Sepsané požadavky zákazníků, ukládané ve vhodném softwarovém nástroji.

Proces analýzy systémových požadavků

Zaměření auditu: Definovaný set funkční a ne-funkčních požadavků. Techniky pro optimalizaci vybraného řešení. Požadavky ověřeny na korektnost a testovatelnost. Dopad systému na jeho provozní okolí. Požadavky utříděny a schváleny. Změny v základních věcech jsou hodnoceny z hlediska dopadu. Systémové požadavky komunikovány všem dotčeným stranám.

Možná zjištění: Analýza požadavků prostřednictvím specializované aplikace.

Proces návrhu architektury systému

Zaměření auditu: Návrh architektury definující jednotlivé elementy. Systémové požadavky uspokojeny. Komunikační interface. Ověřeny požadavky oproti návrhu. Dohledatelnost požadavků vůči architektuře. Architektura komunikována všem dotčeným stranám. Počítáno s lidským faktorem.

Možná zjištění: Zpracovaný návrh systémové architektury.

Proces implementace

V tomto bodu se norma pouze odkazuje na sesterskou normu ISO/IEC 15288 a na svou vlastní podkapitolu 7.1 – Proces implementace softwaru.

Proces integrace

Zaměření auditu: Strategie implementace. Kritéria pro ověření shody s požadavky. Ověření integrace dle kritérií. Strategie pro testování. Integrovaný systém demonstrující shodu s požadavky. Existence funkčních systémových elementů.

Možná zjištění: Softwarové prostředí pro vytvoření spustitelného aplikačního souboru.

Proces testování systému

Zaměření auditu: Nastavená kritéria testování. Integrovaný systém otestován. Záznamy uchovány. Zajištěna připravenost pro předání.

Možná zjištění: Záznamy o testování softwaru.

Proces instalace softwaru

Zaměření auditu: Strategie instalace softwaru. Kritéria pro ověření shody s požadavky. Produkt nainstalován. Zajištěna připravenost produktu.

Možná zjištění: Předávací protokoly z nasazení softwaru v konečném prostředí u zákazníka.

Proces podpory přijetí softwaru

Zaměření auditu: Produkt je zkompleťován a dodán. Testování nabyvatelem. Produkt spuštěn v zákaznickově prostředí. Vzniklé problémy komunikovány k odstranění.

Možná zjištění: Záznamy o testech u zákazníka. Záznamy připomínek a jejich vypořádání.

Proces podpory provozu softwaru

Zaměření auditu: Provozní strategie. Podmínky pro správnou funkčnost softwaru. Software je otestován na správnou funkčnost v zamýšleném prostředí. Software provozován v zamýšleném prostředí. Poskytována pomoc a konzultace zákazníkům dle uzavřené smlouvy.

Možná zjištění: Problematika zodpovědnosti za provoz softwaru definovaná ve smlouvě o dodání. Pokud se nejedná o aplikaci cloudového typu, bývá zpravidla provoz čistě v režii zákazníka.

Proces údržby softwaru

Zaměření auditu: Strategie údržby. Identifikovány změny v systému, funkčnost a interface. Dotyčný software (a jeho dokumentace) je updatován dle potřeby. Upravený software je testován, zda stále uspokojuje požadavky. Upgradované softwary jsou nasazeny u zákazníka. Modifikace softwaru je komunikována všem partnerům.

Možná zjištění: Problematika zodpovědnosti za údržbu softwaru definovaná ve smlouvě o dodání. Pokud se nejedná o aplikaci cloudového typu, bývá zpravidla údržba čistě v režii zákazníka. Případné nezbytné zásahy ze strany dodavatele jsou prováděny buď formou reklamace části systému či další smluvní zakázky.

Proces vyřazení softwaru

Zaměření auditu: Zastavení podpory po určitém čase. Archivace softwaru a dokumentace. Zodpovědnost za zbytkové problémy. Přejechod na nový software. Přístup k archivu dat.

Možná zjištění: Problematika zodpovědnosti za vyřazení softwaru definovaná ve smlouvě o dodání. Pokud se nejedná o aplikaci cloudového typu, bývá zpravidla vyřazení z provozu čistě v režii zákazníka.

Proces implementace softwaru

Zaměření auditu: Strategie implementace softwaru. Technická omezení implementace. Software je zhotoven. Software je zabalen a skladován v souladu s dohodnutými podmínkami.

Možná zjištění: Technická specifikace (příloha smlouvy), obsahující následující typy požadavků: Uživatelské; Požadavky na vstupní data; Systémové požadavky; Matice infrastruktury

Proces analýzy softwarových požadavků

Zaměření auditu: Požadavky vztažené k jednotlivým softwarovým elementům. Požadavky analyzovány na korektnost a ověřitelnost. Analyzován dopad požadavků na provozní okolí. Vztah mezi systémovými a softwarovými požadavky. Postup pro implementaci. Softwarové požadavky schváleny a aktualizovány. Změny požadavků analyzovány na dopad v nákladech, termínu atd. Softwarové požadavky komunikovány všem zúčastněným stranám.

Možná zjištění: Viz Proces implementace softwaru.

Proces návrhu softwarové architektury

Zaměření auditu: Softwarová architektura. Interní a externí interface. Konzistentnost a dohledatelnost mezi požadavky a designem.

Možná zjištění: Vytvořený návrh architektury systému. Např. databázový server, task server (služba, která vykonává všechny operace nad daty, poskytuje rozhraní pro komunikaci s dalšími systémy třetích stran), server pro aktualizace klientských aplikací, klientské aplikace.

Proces detailního návrhu softwaru

Zaměření auditu: Detailní design všech softwarových komponent. Externí interface pro všechny komponenty. Soulad a dohledatelnost mezi designem, požadavky a softwarovou architekturou.

Možná zjištění: Vytvořen detailních návrh systému, rozpracování jednotlivých subsystémů navržených v architektuře.

Proces tvorby softwaru

Zaměření auditu: Verifikační kritéria. Softwarové komponenty vytvořeny dle návrhu. Konzistentnost a dohledatelnost mezi požadavky a komponentami. Ověření komponent oproti stanoveným kritériím.

Možná zjištění: Tvorba jednotlivých subsystémů, záznamy ověření jejich funkčnosti.

Proces integrace softwaru

Zaměření auditu: Integrační strategie. Verifikační kritéria. Ověření dle verifikačních kritérií. Softwarové komponenty vytvořeny dle specifikace. Archivované výsledky integračních testů. Konzistentnost a dohledatelnost mezi komponenty a designem. Návratová strategie. (Strategie pro případnou potřebu zpětného downgradu.)

Možná zjištění: Propojení všech subsystémů. Záznamy o integraci.

Proces testování softwaru

Zaměření auditu: Kritéria pro stanovení souladu s požadavky. Integrovaný software otestován dle kritérií. Záznamy o testování uchovány. Návratová strategie.

Možná zjištění: Nastavená testovací kritéria, záznamy o provedení testů, výsledky testování.

Proces řízení softwarové dokumentace

Zaměření auditu: Strategie pro identifikaci a tvorbu potřebné dokumentace. Standardy pro vývoj softwaru. Vyvinutá dokumentace. Obsah všech dokumentů specifikován, přezkoumán a schválen. Dokumentace udržována dle specifikovaných kritérií.

Možná zjištění: Vytvořená softwarová dokumentace. Vývojová prostředí umožňující automatickou generaci dokumentace.

Proces řízení konfigurace softwaru

Zaměření auditu: Strategie řízení konfigurace softwaru. Definované výstupy z procesu/projektu. Řízení změn a uvolnění produktů. Změny a uvolnění produktů zpřístupněny dotčeným stranám. Stav produktů a jejich změn zaznamenáván. Zajištěna úplnost a konzistentnost předmětů. Skladování, zacházení a doručování je řízeno.

Možná zjištění: Využívání nástrojů umožňujících jednoznačnou identifikaci konfiguračních položek.

Proces zajištění kvality softwaru

Zaměření auditu: Strategie zajištění kvality. Důkazy, že je kvalita softwaru zajišťována. Nedostatky a neshody identifikovány a zaznamenány. Ověření shody produktů, procesů a aktivit se standardy.

Možná zjištění: Požadavky na zajištění kvality jsou dány požadavky ze strany klíčových zákazníků a podmínkou vybraných typů certifikace, např. UCL.

Proces verifikace softwaru

Zaměření auditu: Strategie verifikace. Kritéria pro verifikaci. Verifikace provedena. Defekty identifikovány a zaznamenány. Výsledky verifikace zpřístupněny zákazníkovi a ostatním zapojeným stranám.

Možná zjištění: Nastavená verifikační kritéria. Záznamy o provedené verifikaci. Záznamy o výsledcích verifikace.

Proces validace softwaru

Zaměření auditu: Strategie validace. Kritéria pro validaci. Validace provedena. Problémy identifikovány a zaznamenány. Důkazy, že je vyvinutý software vhodný pro jeho zamýšlené využití. Výsledky validace zpřístupněny zákazníkovi a ostatním zapojeným stranám.

Možná zjištění: Nastavená validační kritéria. Záznamy o provedené validaci. Záznamy o výsledcích validace.

Proces přezkoumání softwaru

Zaměření auditu: Pravidelná technická přezkoumání a přezkoumání managementu. Status a produkt aktivit vyhodnocen při každém přezkoumání. Výsledky přezkoumání zveřejněny všem dotčeným stranám. Nápravné akce z přezkoumání sledovány. Rizika a problémy identifikovány a zaznamenány.

Možná zjištění: Přezkoumání softwaru jako součást projektového řízení. Zápisy z přezkoumání.

Proces auditu softwaru

Zaměření auditu: Strategie auditu. Posouzení shody softwaru, služeb a procesů dle požadavků, plánů a dohod. Audity provedeny vhodnou nezávislou stranou. Problémy detekované během auditu jsou zaznamenány a komunikovány k osobám zodpovědným za nápravu.

Možná zjištění: Zpráva z auditu.

Proces řešení softwarových problémů

Zaměření auditu: Strategie řešení problémů. Problémy zaznamenány, identifikovány a zařazeny. Problémy analyzovány a jsou navržena vhodná řešení. Řešení problémů implementována. Problémy sledovány až do úplného vyřešení. Status všech reportovaných problémů je znám.

Možná zjištění: Záznamy o vzniklých problémech. Soupis aktuálních problémů.

Proces doménového inženýrství

Zaměření auditu: Reprezentační formy doménových modelů a doménových architektur. Hranice domény a její vztahy s ostatními doménami. Doménový model zachycující specifiky dané domény. Doménová architektura popisující rodinu systémů v doméně včetně jejich specifikací. Určena aktiva spadající do domény. Aktiva spadající do domény zajištěna nebo vyvinuta a udržována po celou dobu životního cyklu. Doménové modely a architektury udržovány po celou dobu životního cyklu.

Proces řízení opětovného použití aktiv

Zaměření auditu: Strategie řízení aktiv. Schéma klasifikace aktiv. Kritéria pro přijetí aktiv, certifikaci a vyřazení. Systém uchovávání a vyzvedávání aktiv. Využívání a změny aktiv monitorovány. Uživatelé aktiv upozorněny na problémy, modifikace, nové verze a vyřazení daných aktiv.

Proces řízení programu opětovného použití

Zaměření auditu: Strategie opětovného použití. Domény potenciálního opětovného použití stanoveny. Hodnocení schopnosti organizace opětovně využívat aktiva. Hodnocení potenciálu opětovného použití všech domén. Návrhy na opětovné využití evaluovány. Strategie opětovného použití implementována do organizace. Zpětná vazba, komunikace a oznamování mezi zapojenými stranami. Program opětovného použití je monitorován a hodnocen.

5.5 Finalizace

Posledním krokem ve vytváření check listu pak již byla pouze formalizace všech výše uvedených kroků. Došlo tak k vytvoření tabulky dle kritérií stanovených v bodu 5.3 a jejímu naplnění údaji získanými v bodě 5.4.

5.6 Vyhodnocení zpracovaného řešení

Check listy jsou ve společnosti dlouhodobě využívány jako efektivní pomůcka auditora. Jsou vytvářeny ke všem normám, dle kterých aktivně probíhají certifikační procesy. Mnou zpracované řešení tak navazuje na dlouhodobou tradici v této oblasti.

Jeho vnitřní struktura také navazuje na úspěšně používané check listy vytvořené pro certifikaci dle jiných norem. Její omezení pouze na výstupy jednotlivých procesů je do určité míry inovativní. Důvodem byla nereálnost praktického ověření provádění veškerých stanovených aktivit v rámci všech nastavených procesů.

Vytvořený check list bude předán do vnitropodnikového informačního systému, kde bude k dispozici všem zainteresovaným stranám k pouhému nahlédnutí i k aktivnímu využívání. Míra jeho skutečného využívání pak závisí především na poptávce po certifikaci softwaru dle normy ISO/IEC 12007 ze strany koncových zákazníků.

6 Závěr

Finálnímu zpracování vytvořeného check listu předcházela průběžná kvalitní analýza situace a možností řešení. Při přemýšlení nad zadaným problémem jsem vycházel jak ze svých zkušeností a nápadů, tak z konzultací s kolegy z auditorského týmu společnosti EZÚ, s.p.

Všechny vykonávané činnosti byly důsledně plánovány a průběžně vyhodnocovány pro dosažení nejlepšího možného výsledku. Celková tvorba byla rozdělena do jednotlivých dílčích aktivit a ty byly dále naplánovány rovnoměrně mezi ostatní činnosti vykonávané v rámci mé roční řízené praxe tak, aby s nimi byly v souladu a vzájemně se doplňovaly. Například jsem tak využil možnosti pobavit se v rámci auditu v jedné z předních světových firem s člověkem zodpovědným za plánování vývojových činností pro vestavěný software v dopravních prostředcích, na který byla kladena podmínka úspěšné certifikace právě dle normy ISO/IEC 12207.

Všechny aktivity byly realizovány se zřetelem na společný cíl, jímž bylo vytvoření funkčního check listu pro potřeby společnosti. Tyto na sebe logicky navazovaly a přispívaly tak ke zdárnému splnění zadaného úkolu. Celý problém tak byl zpracován s vysokou efektivitou.

Hlavním cílem této bakalářské práce bylo vytvoření nějaké pomůcky či metodického postupu, který by umožnil provádění certifikace softwaru dle normy ISO/IEC 12007. Tohoto cíle bylo bezesporu dosaženo. Vytvořený check list byl předán do informačního systému, kde bude k dispozici všem zapojeným stranám.

Praktickým vytvořením tohoto check listu tak došlo k odstranění zásadní překážky, která bránila společnosti rozšířit své portfolio certifikačních činností o tuto oblast.

7 Seznam citací

ČSN ISO/IEC 12207. *Informační technologie: Procesy v životním cyklu softwaru*. Praha: ČESKÝ NORMALIZAČNÍ INSTITUT, 1997, str. 5

Audit. *Auditor Dictionary* [online]. [cit. 2014-04-14]. Dostupné z: <http://www.projectauditors.com/Dictionary2/1.8/index.php/term/,62555c9cae53606f68555ca75a.xhtml>

Bell, T.E. a Thayer, T.A. *Software requirements: Are they really a problem?* [online]. 1976 [cit. 2014-03-31]. Dostupné z http://pdf.aminer.org/000/361/405/software_requirements_are_they_really_a_problem.pdf

BIRENDRA, P. a KAUR, A. *A Proposal for New Software Development Life Cycle Model: PARTH's Model*. [online] Singapore: Research Publishing Services, 2010 ProQuest Technology Collection. Dostupné z: http://dx.doi.org/10.3850/978-981-08-7618-0_1352

BOEHM, B. ACM SIGSOFT *Software Engineering Notes* [online]. 1986-08-01, vol. 11, issue 4, s. 22-42 [cit. 2014-03-31]. DOI: 10.1145/12944.12948. Dostupné z: <http://portal.acm.org/citation.cfm?doid=12944.12948>

BROOKS, F. *Mystical man-month - essays on software engineering*. Vyd. 1. Massachusetts: Addison-Wesley, 1982, 195 s. ISBN 02-010-0650-2.

Český institut pro akreditaci, o.p.s.: O nás. *Český institut pro akreditaci, o.p.s.* [online]. [cit. 2013-12-13]. Dostupné z: <http://www.cia.cz/o-nas.aspx>

ČSN EN ISO/IEC 17021:2011. *Posuzování shody: Požadavky na orgány poskytující služby auditů a certifikace systémů managementu*. 1. vyd. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011, s. 11.

DOD-STD-2167, *MILITARY STANDARD: DEFENSE SYSTEM SOFTWARE DEVELOPMENT* [online] 1985-06-04, [cit. 2014-03-31] Dostupné z: http://www.everyspec.com/DoD/DoD-STD/DOD-STD-2167_278/

ISO/IEC 12207:2008. *Systems and software engineering: Software life cycle processes*. Second edition. United States of America: ISO/IEC-IEEE, 2008, s. 1.

MANDAL, S., KANDAR, S. a RAY, P. *Open Incremental Model- A Open Source Software Development Life Cycle Model (OSDLC)*. [online] International Journal of Computer Applications, 2011, vol. 21, no. 1 ProQuest Technology Collection. ISSN 09758887. Dostupné z: <http://dx.doi.org/10.5120/2473-3327>.

Manifest agilního vývoje software. [online]. [cit. 2014-03-31]. Dostupné z: <http://agilemanifesto.org/iso/cs/>

Open Source Initiative. *The Open Source Definition* [online]. [cit. 2014-04-14]. Dostupné z: <http://opensource.org/osd>

Principy stojící za Agilním Manifestem. [online]. [cit. 2014-04-15]. Dostupné z: <http://agilemanifesto.org/iso/cs/principles.html>

Spiral model. *SDLC Tutorials* [online]. [cit. 2014-02-10]. Dostupné z: <http://www.sdgc.ws/spiral-model/>

Symposium on advanced programming methods for digital computers. Washington D.C., USA, 1956.

The Free Dictionary [online]. 2012 [cit. 2013-12-16]. Dostupné z: <http://www.thefreedictionary.com/computer+software>

VARIA, Vishal. *ASSIGNMENT ON SOFTWARE, ITS TYPES, SYSTEM SOFTWARE, OPERATING SYSTEM, FUNCTION & ITS TYPES*[online]. Rajkot, 2011 [cit. 2013-12-16]. Dostupné z: http://www.academia.edu/4564920/ASSIGNMENT_ON_SOFTWARE_ITS_TYPES_SYSTEM_SOFTWARE_OPERATING_SYSTEM_FUNCTION_and_ITS_TYPES. Assignment. Department of Education, Saurashtra University.

What is SDLC. *SDLC Tutorials* [online]. [cit. 2014-01-02]. Dostupné z: <http://www.sdgc.ws/what-is-sdgc/>

WINSTON, Royce. *Managing the development of large software systems*. [online]. 1970
[cit. 2014-03-31]. Dostupné z:
<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>

8 Bibliografie

BRUCKNER, Tomáš. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012, 357 s. Management v informační společnosti. ISBN 978-80-247-4153-6.

Český institut pro akreditaci, o.p.s.: O nás. *Český institut pro akreditaci, o.p.s.* [online]. [cit. 2013-12-13]. Dostupné z: <http://www.cia.cz/o-nas.aspx>

ČSN EN ISO/IEC 17021:2011. *Posuzování shody: Požadavky na orgány poskytující služby auditů a certifikace systémů managementu*. 1. vyd. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika*. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009, 496 s. Expert (Grada). ISBN 978-80-247-2615-1.

ISO/IEC 12207:2008. *Systems and software engineering: Software life cycle processes*. Second edition. United States of America: ISO/IEC-IEEE, 2008.

Open Projects. *Open Projects* [online]. [cit. 2014-04-14]. Dostupné z: <http://www.openprojects.org/software-definition.htm>

SDLC Tutorials [online]. 2012-04-15 [cit. 2014-01-04]. Dostupné z: <http://www.sdlic.ws/>

The Free Dictionary [online]. 2012 [cit. 2013-12-16]. Dostupné z: <http://www.thefreedictionary.com/computer+software>

VARIA, Vishal. *ASSIGNMENT ON SOFTWARE, ITS TYPES, SYSTEM SOFTWARE, OPERATING SYSTEM, FUNCTION & ITS TYPES* [online]. Assignment. Department of Education, Saurashtra University. Rajkot, 2011 [cit. 2013-12-16]. Dostupné z: http://www.academia.edu/4564920/ASSIGNMENT_ON_SOFTWARE_ITS_TYPES_SYSTEM_SOFTWARE_OPERATING_SYSTEM_FUNCTION_and_ITS_TYPES.

9 Seznam příloh

Příloha A – Vypracovaný check list pro provádění auditů dle normy ISO/IEC 12207

Číslo článku	Název článku	Zaměření auditu	Možná zjištění
6	Procesy životního cyklu softwaru		
6.1	Smluvní procesy		
6.1.1	Proces akvizice	Potřeby a cíle akvizice; Cíle a kritéria přijetí akvizice. Dohoda/smlouva jasně stanovující postavení, očekávání a odpovědnosti obou stran. Vybrán dodavatel a dodán produkt uspokojující potřeby. Byly dodrženy stanovené náklady/rozpočet atd.	V průběhu jednání se zákazníkem vzniká technická specifikace zakázky, která je součástí finální nabídky.
6.1.2	Proces dodání	Identifikován nabyvatel produktu. Odpověď na požadavek/poptávku nabyvatele. Dohoda o dodání produktu. Produkt uspokojuje schválené požadavky. Produkt je dodán a nainstalován dle schválených požadavků.	Proces dodání je jasně vymezen v uzavírané smlouvě.
6.2	Procesy přípravné fáze projektů		
6.2.1	Proces řízení modelu životního cyklu	Politiky a postupy pro řízení životního cyklu SW. Definovány odpovědnosti a pravomoci. Procesy, modely a postupy životního cyklu jsou definovány, udržovány a zlepšovány.	Průběh procesů životního cyklu návrhu, implementace a udržování dodávaného SW produktu je naplánován a odsouhlasen podle uznávaných standardů v rámci smluv s dodavateli.
6.2.2	Proces řízení infrastruktury	Požadavky na podporu procesů infrastrukturou. Definovány a specifikovány elementy infrastruktury. Tyto elementy jsou zajištěny a implementovány. Je udržována a zlepšována stabilní a spolehlivá infrastruktura.	Hardwarová i softwarová infrastruktura je jasně definována. Seznam používaných aplikací.
6.2.3	Proces řízení projektového portfolia	Obchodní příležitosti, investice a potřeby. Zdroje pro každý projekt. Zodpovědnost za projekty stanovena. Projekty splňující kritéria udržovány.	Management projektů probíhá prostřednictvím plánování, dokladování a přezkoumávání pomocí některé z aplikací pro projektové řízení.

Projekty nesplňující kritéria ukončeny.			
6.2.4	Proces řízení lidských zdrojů	Identifikovány dovednosti vyžadované projektem. Zajištěny lidské zdroje pro projekty. Dovednosti zaměstnanců udržovány a rozvíjeny. Řešeny konflikty lidských zdrojů mezi projekty. Individuální znalosti a dovednosti jsou shromažďovány a využívány napříč organizací.	Jsou identifikovány požadované dovednosti pro funkce a dílčí pozice. Záznamy o alokaci lidských zdrojů k jednotlivým projektům.
6.2.5	Proces řízení kvality	Politiky a postupy řízení kvality. Cíle řízení kvality. Zodpovědnost za řízení kvality. Spokojenost zákazníků hodnocena. Opatření v případě, že cíle kvality nejsou plněny.	Certifikace dle ISO 9001. Příručka jakosti. Podnikové cíle v oblasti jakosti.
6.3	Projektové procesy		
6.3.1	Proces plánování projektu	Stanoven rozsah projektu. Vyhodnocena schopnost dosažení cílů. Úkoly a zdroje odhadnuty. Interface pro vnitřní i vnější komunikaci. Plán realizace projektu.	Management projektů probíhá prostřednictvím plánování, dokladování a přezkoumávání pomocí některé z aplikací pro projektové řízení.
6.3.2	Proces přezkoumání a kontroly	Postup projektu monitorován. Interface pro komunikaci monitorován. Podniknuty kroky pro odstranění odchylek od plánu. Dosahovány a zaznamenávány cíle projektu.	Záznamy z porad, přezkoumání projektů managementem.
6.3.3	Proces řízení rozhodování	Strategie řízení rozhodování. Definovány alternativní možnosti řešení. Vybrána nejvhodnější alternativa. Dokumentace o rozhodnutí a argumentech pro ně.	
6.3.4	Proces řízení rizik	Stanoven rozsah řízení rizik. Přiměřené strategie řízení rizik. Průběžná identifikace a analýza rizik. Nastaveny metriky pro hodnocení rizik. Přijata vhodná opatření k odstranění rizik.	Rizikové analýzy projektů, SWOT analýzy.

6.3.5	Proces řízení nastavení	Strategie řízení nastavení. Definovány položky vyžadující řízení nastavení. Pravidla (zásady) nastavení. Monitoring změn předmětů podléhajících řízení nastavení. Kontrola nastavení uvolněných předmětů.	Procesní příručka, manuály pro konfiguraci.
6.3.6	Proces řízení informací	Identifikovány informace k řízení. Forma interpretace informací. Požadavky na převod a odstranění informací. Záznamy o stavu informací. Informace musí být aktuální, úplné a správné. Informace zpřístupněny správným osobám.	Zavedená aplikace pro sdílení informací, aplikace pro řízení projektů se sdílenými přístupy, firemní intranet.
6.3.7	Proces hodnocení	Identifikovány informační potřeby řídicích procesů. Vhodná sada metrik pro hodnocení. Plánování hodnocení. Požadovaná data posbírána, analyzována a uchována. Hodnotící procesy jsou evaluovány. Zlepšení jsou komunikována k majiteli hodnotícího procesu.	Záznamy o hodnocení. Nastavené hodnotící procesy a oblasti v rámci používané aplikace pro projektové řízení.
6.4	Technické procesy		
6.4.1	Proces definice požadavků zainteresovaných stran	Určen kontext a charakteristiky použití služby. Omezení systémového řešení. Požadavky dohledatelné. Zásady definování systémových požadavků. Zásady pro ověření shody. Zásady pro vyjednávání o dodání.	Sepsané požadavky zákazníků, ukládané ve vhodném softwarovém nástroji.
6.4.2	Proces analýzy systémových požadavků	Definovaný set funkční a ne-funkčních požadavků. Techniky pro optimalizaci vybraného řešení. Požadavky ověřeny na korektnost a testovatelnost. Dopad systému na jeho provozní okolí. Požadavky utříděny a schváleny. Změny v základních věcech jsou hodnoceny z hlediska	Analýza požadavků prostřednictvím specializované aplikace.

		dopadu. Systémové požadavky komunikovány všem dotčeným stranám.	
6.4.3	Proces návrhu architektury systému	Návrh architektury definující jednotlivé elementy. Systémové požadavky uspokojeny. Komunikační interface. Ověřeny požadavky oproti návrhu. Dohledatelnost požadavků vůči architektuře. Architektura komunikována všem dotčeným stranám. Počítáno s lidským faktorem.	Zpracovaný návrh systémové architektury.
6.4.4	Proces implementace		
6.4.5	Proces integrace	Strategie implementace. Kritéria pro ověření shody s požadavky. Ověření integrace dle kritérií. Strategie pro testování. Integrovaný systém demonstrující shodu s požadavky. Existence funkčních systémových elementů.	Softwarové prostředí pro vytvoření spustitelného aplikačního souboru.
6.4.6	Proces testování systému	Nastavená kritéria testování. Integrovaný systém otestován. Záznamy uchovány. Zajištěna připravenost pro předání.	Záznamy o testování softwaru.
6.4.7	Proces instalace softwaru	Strategie instalace softwaru. Kritéria pro ověření shody s požadavky. Produkt nainstalován. Zajištěna připravenost produktu.	Předávací protokoly z nasazení softwaru v konečném prostředí u zákazníka.
6.4.8	Proces podpory přijetí softwaru	Produkt je zkompletován a dodán. Testování nabyvatelem. Produkt spuštěn v zákaznickově prostředí. Vzniklé problémy komunikovány k odstranění.	Záznamy o testech u zákazníka. Záznamy připomínek a jejich vypořádání.
6.4.9	Proces provozu softwaru	Provozní strategie. Podmínky pro správnou funkčnost softwaru. Software je otestován na správnou funkčnost	Problematika zodpovědnosti za provoz softwaru definovaná ve smlouvě o dodání.

		<p>v zamýšleném prostředí.</p> <p>Software provozován v zamýšleném prostředí.</p> <p>Poskytována pomoc a konzultace zákazníkům dle uzavřené smlouvy.</p>	
6.4.10	Proces údržby softwaru	<p>Strategie údržby.</p> <p>Identifikovány změny v systému, funkčnost a interface.</p> <p>Dotyčný software (a jeho dokumentace) je updatován dle potřeby.</p> <p>Upravený software je testován, zda stále uspokojuje požadavky.</p> <p>Upgradované softwary jsou nasazeny u zákazníka.</p> <p>Modifikace softwaru je komunikována všem partnerům.</p>	Problematika zodpovědnosti za údržbu softwaru definovaná ve smlouvě o dodání.
6.4.11	Proces vyřazení softwaru	<p>Zastavení podpory po určitém čase.</p> <p>Archivace softwaru a dokumentace.</p> <p>Zodpovědnost za zbytkové problémy.</p> <p>Přechod na nový software.</p> <p>Přístup k archivu dat.</p>	Problematika zodpovědnosti za vyřazení softwaru definovaná ve smlouvě o dodání.
7	Softwarově specifické procesy		
7.1	Procesy implementace softwaru		
7.1.1	Proces implementace softwaru	<p>Strategie implementace softwaru.</p> <p>Technická omezení implementace.</p> <p>Software je zhotoven.</p> <p>Software je zabalen a skladován v souladu s dohodnutými podmínkami.</p>	<p>Technická specifikace (příloha smlouvy), obsahující následující typy požadavků:</p> <ul style="list-style-type: none"> - Uživatelské - Požadavky na vstupní data - Systémové požadavky - Matice infrastruktury
7.1.2	Proces analýzy softwarových požadavků	<p>Požadavky vztažené k jednotlivým softwarovým elementům.</p> <p>Požadavky analyzovány na korektnost a ověřitelnost.</p> <p>Analyzován dopad požadavků na provozní okolí.</p> <p>Vztah mezi systémovými a softwarovými požadavky.</p> <p>Postup pro implementaci.</p>	Viz 7.1.1

		Softwarové požadavky schváleny a aktualizovány. Změny požadavků analyzovány na dopad v nákladech, termínu atd. Softwarové požadavky komunikovány všem zúčastněným stranám.	
7.1.3	Proces návrhu softwarové architektury	Softwarová architektura. Interní a externí interface. Konzistentnost a dohledatelnost mezi požadavky a designem.	Vytvořený návrh architektury systému. Např. databázový server, task server (služba, která vykonává všechny operace nad daty, poskytuje rozhraní pro komunikaci s dalšími systémy třetích stran), server pro aktualizace klientských aplikací, klientské aplikace.
7.1.4	Proces detailního návrhu softwaru	Detailní design všech softwarových komponent. Externí interface pro všechny komponenty. Soulad a dohledatelnost mezi designem, požadavky a softwarovou architekturou.	Vytvořen detailní návrh systému, rozpracování jednotlivých subsystémů navržených v architektuře.
7.1.5	Proces tvorby softwaru	Verifikační kritéria. Softwarové komponenty vytvořeny dle návrhu. Konzistentnost a dohledatelnost mezi požadavky a komponentami. Ověření komponent oproti stanoveným kritériím.	Tvorba jednotlivých subsystémů, záznamy ověření jejich funkčnosti.
7.1.6	Proces integrace softwaru	Integrační strategie. Verifikační kritéria. Ověření dle verifikačních kritérií. Softwarové komponenty vytvořeny dle specifikace. Archivované výsledky integračních testů. Konzistentnost a dohledatelnost mezi komponenty a designem. Návratová strategie. (Strategie pro případnou potřebu zpětného downgradu.)	Propojení všech subsystémů. Záznamy o integraci.
7.1.7	Proces testování softwaru	Kritéria pro stanovení souladu s požadavky. Integrovaný software otestován dle kritérií. Záznamy o testování uchovány. Návratová strategie.	Nastavená testovací kritéria, záznamy o provedení testů, výsledky testování.

7.2	Podpůrné procesy		
7.2.1	Proces řízení softwarové dokumentace	Strategie pro identifikaci a tvorbu potřebné dokumentace. Standardy pro vývoj softwaru. Vyvinutá dokumentace. Obsah všech dokumentů specifikován, přezkoumán a schválen. Dokumentace udržována dle specifikovaných kritérií.	Vytvořená softwarová dokumentace. Vývojová prostředí umožňující automatickou generaci dokumentace.
7.2.2	Proces řízení konfigurace softwaru	Strategie řízení konfigurace softwaru. Definované výstupy z procesu/projektu. Řízení změn a uvolnění produktů. Změny a uvolnění produktů zpřístupněny dotčeným stranám. Stav produktů a jejich změn zaznamenáván. Zajištěna úplnost a konzistentnost předmětů. Skladování, zacházení a doručování je řízeno.	Využívání nástrojů umožňujících jednoznačnou identifikaci konfiguračních položek.
7.2.3	Proces zajištění kvality softwaru	Strategie zajištění kvality. Důkazy, že je kvalita softwaru zajišťována. Nedostatky a neshody identifikovány a zaznamenány. Ověření shody produktů, procesů a aktivit se standardy.	Požadavky na zajištění kvality jsou dány požadavky ze strany klíčových zákazníků a podmínkou vybraných typů certifikace, např. UCL.
7.2.4	Proces verifikace softwaru	Strategie verifikace. Kritéria pro verifikaci. Verifikace provedena. Defekty identifikovány a zaznamenány. Výsledky verifikace zpřístupněny zákazníkovi a ostatním zapojeným stranám.	Nastavená verifikační kritéria. Záznamy o provedené verifikaci. Záznamy o výsledcích verifikace.
7.2.5	Proces validace softwaru	Strategie validace. Kritéria pro validaci. Validace provedena. Problémy identifikovány a zaznamenány. Důkazy, že je vyvinutý software vhodný pro jeho zamýšlené využití.	Nastavená validační kritéria. Záznamy o provedené validaci. Záznamy o výsledcích validace.

		Výsledky validace zpřístupněny zákazníkovi a ostatním zapojeným stranám.	
7.2.6	Proces přezkoumání softwaru	Pravidelná technická přezkoumání a přezkoumání managementu. Status a produkt aktivit vyhodnocen při každém přezkoumání. Výsledky přezkoumání zveřejněny všem dotčeným stranám. Nápravné akce z přezkoumání sledovány. Rizika a problémy identifikovány a zaznamenány.	Přezkoumání softwaru jako součást projektového řízení. Zápisy z přezkoumání.
7.2.7	Proces auditu softwaru	Strategie auditu. Posouzení shody softwaru, služeb a procesů dle požadavků, plánů a dohod. Audity provedeny vhodnou nezávislou stranou. Problémy detekované během auditu jsou zaznamenány a komunikované k osobám zodpovědným za nápravu.	Zpráva z auditu.
7.2.8	Proces řešení softwarových problémů	Strategie řešení problémů. Problémy zaznamenány, identifikovány a zařazeny. Problémy analyzovány a jsou navržena vhodná řešení. Řešení problémů implementována. Problémy sledovány až do úplného vyřešení. Status všech reportovaných problémů je znám.	Záznamy o vzniklých problémech. Soupis aktuálních problémů.
7.3	Procesy opětovného využití		Systém monitoringu aktiv a záznamy o jeho aktivním využívání.
7.3.1	Proces doménového inženýrství	Reprezentační formy doménových modelů a doménových architektur. Hranice domény a její vztahy s ostatními doménami. Doménový model zachycující specifika dané domény. Doménová architektura popisující rodinu systémů v doméně včetně jejich specifikací. Určena aktiva spadající do domény. Aktiva spadající do domény zajištěna nebo vyvinuta a udržována po celou dobu životního cyklu.	

		Doménové modely a architektury udržovány po celou dobu životního cyklu.
7.3.2	Proces řízení opětovného použití aktiv	<p>Strategie řízení aktiv. Schéma klasifikace aktiv. Kritéria pro přijetí aktiv, certifikaci a vyřazení. Systém uchovávání a vyzvedávání aktiv. Využívání a změny aktiv monitorovány. Uživatelé aktiv upozorněny na problémy, modifikace, nové verze a vyřazení daných aktiv.</p>
7.3.3	Proces řízení programu opětovného použití	<p>Strategie opětovného použití. Domény potenciálního opětovného použití stanoveny. Hodnocení schopnosti organizace opětovně využívat aktiva. Hodnocení potenciálu opětovného použití všech domén. Návrhy na opětovné využití evaluovány. Strategie opětovného použití implementována do organizace. Zpětná vazba, komunikace a oznamování mezi zapojenými stranami. Program opětovného použití je monitorován a hodnocen.</p>